

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 11 Gennaio 2022 – Versione C**

**Tempo a disposizione: 180 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>C1.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>C2.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 2
<b>C3.zip</b>	file zip contenente pagine Web e codice React.js per punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto.**

**N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.**

---

**ESERCIZIO 1 (11 punti)**

Si realizzi una applicazione Web per la **prenotazione di campi da tennis**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere ai soli utenti autenticati di accedere ad una pagina per la prenotazione di un campo da tennis: ogni richiesta di prenotazione deve specificare numero del campo (da 1 a 6), giorno (da 1 a 365) e orario di inizio (da 0 a 23). La correttezza dei dati di ingresso deve essere controllata lato cliente prima dell'invio al servitore.

Una volta inseriti tutti i dati correttamente, l'applicazione Web deve automaticamente (senza pressione esplicita di alcun bottone da parte del cliente) invocare un componente server-side che si deve occupare di verificare se la richiesta di prenotazione è possibile. In particolare:

- Se il campo è già occupato in quello slot, la prenotazione non avrà luogo e dovrà essere inviato un messaggio di errore al cliente, mostrandogli 5 slot liberi di qualche campo nella stessa giornata, ove ce ne siano; questo risultato deve essere inviato in formato JSON;
- Se il campo è libero in quello slot, la prenotazione va accettata, ma solo **in modo temporaneo**; infatti la prenotazione sarà definitiva solo quando un altro utente avrà confermato di essere l'altro tennista a giocare in quello slot (vedi punto seguente);
- Se il campo è soggetto in quello slot a **prenotazione in stato temporaneo**, allora con questa richiesta la prenotazione diventerà definitiva e i due tennisti dovranno essere notificati del completamento della prenotazione il prima possibile.

Si faccia inoltre in modo che le **prenotazioni temporanee non diventate definitive entro 2 ore siano annullate**, con conseguente notifica del tennista il prima possibile.

Gestione: si preveda inoltre che un amministratore di sistema, previa autenticazione, possa avere accesso a una pagina **admin.jsp** in cui sono elencate tutte le prenotazioni in stato temporaneo e in cui si possa forzare il cambiamento di stato di ciascuna di esse verso lo stato di "annullato".

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 11 Gennaio 2022 – Versione C**

**ESERCIZIO 2 (11 punti)**

Si realizzi una applicazione Web per **determinare se una matrice  $4 * 4$  rappresenta un quadrato magico**. Si definisce “quadrato magico” una qualunque matrice quadrata di numeri naturali in cui sono rispettate le seguenti proprietà: non ci possono essere elementi ripetuti; la somma di ogni riga è uguale alla somma di ogni colonna ed è uguale alla somma di ciascuna delle due diagonali. L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere all'utente di inserire i 16 elementi della matrice considerata, controllando che siano effettivamente numeri naturali e non siano ripetuti. Solo dopo avere terminato l'inserimento di tutti gli elementi validi, l'applicazione deve invocare automaticamente l'esecuzione server-side dell'operazione concorrente di determinazione del quadrato magico: in modo completamente concorrente, una **JSP J1** dovrà controllare che le somme di ogni riga siano uguali; una **servlet S1** dovrà controllare che le somme di ogni colonna siano uguali; infine una **JSP J1** dovrà controllare che le somme delle due diagonali siano uguali. Ogni componente server-side dovrà restituire in modo autonomo e **in formato JSON** un valore “vero”/”falso” a seconda del risultato di tale controllo e, nel caso di “vero”, anche il risultato della somma.

Solo dopo avere ricevuto tutti e tre i risultati, il cliente dovrà visualizzare il risultato complessivo, ovvero se gli elementi inseriti appartengono a un quadrato magico oppure no.

Infine, si realizzi una *estensione* (versione 2) dell'applicazione descritta sopra che **impedisca a un utente di usufruire più di 10 richieste di servizio al giorno**.

---

**ESERCIZIO 3 (11 punti)**

Si realizzi in React un'applicazione Web lato cliente che simuli il gioco “**Campo Minato**”. L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione.** In tale sezione sono presenti due elementi di input per l'inserimento, rispettivamente, dell'altezza ( $a > 5$ ) e della larghezza ( $l > 5$ ) della griglia di gioco. Solo dopo avere acquisito questi dati, l'utente potrà iniziare a giocare.
- **Sezione Gioco.** Tale sezione deve contenere una griglia rettangolare ( $a * l$ ) di celle inizialmente tutte di colore grigio, ciascuna delle quali può potenzialmente nascondere una mina. Per ogni riga della griglia, vengono posizionate 2 mine su altrettante celle scelte in modo del tutto casuale. L'utente deve percorrere 5 passi sulla griglia con l'obiettivo di non calpestare alcuna mina. Ogni passo viene simulato da un click sulla cella che l'utente intende calpestare. Quando l'utente effettua un passo su una cella, su di essa appare il numero corrispondente al passo effettuato. Inoltre, se la cella calpestata non contiene una mina, essa si colora di blu; altrimenti si colora di rosso. Se in una partita l'utente calpesta per tre volte una cella che nasconde una mina, gli appare una finestra di dialogo con la scritta “Partita terminata: hai perso!”. Se invece l'utente termina i 5 passi senza mai calpestare una mina, viene visualizzata una finestra di dialogo con la scritta “Partita terminata: hai vinto!”. Dismessa la finestra di dialogo, la griglia di gioco viene resettata (mantenendo le stesse dimensioni acquisite in fase di configurazione) e l'utente può cominciare una nuova partita.
- **Sezione Conteggio.** In questa sezione, occorre visualizzare due contatori che tengono traccia, rispettivamente, del numero di partite vinte e del numero di partite perse dall'utente.